Steps Taken to Install Spinnaker on Kubernetes Cluster

1. Prerequisites
    a. Have a Kubernetes cluster with enough resources to handle spinnaker
    b. A docker registry (docker hub, ECR, etc)
    c. Persistent storage (S3, Minio, etc)
    d. An account with permissions to read from docker registry
    e. Account with permissions to use persistent storage
    f. Make a decision in regards to what kinds of artifacts you will use
    g. Account(s) with permissions for those artifacts (might not be necessary, depends on what kind of artifacts you want to use)

2. Connect to Kubernetes Cluster
    a. //Run these commands from bash/linux/macOS environment
    aws configure
    aws eks --region <REGION> update-kubeconfig --name<NAME> --profile default

3. Create Kubernetes Objects
    a. Create spinacct.yaml

```
apiVersion: v1
kind: ServiceAccount
metadata:
  name: spinnaker-service-account
  namespace: default
---
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: spinnaker-role-binding
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: cluster-admin
subjects:
- namespace: default
  kind: ServiceAccount
```

name: spinnaker-service-account

b. Create the account

kubectl create -f spinacct.yaml

serviceaccount "spinnaker-service-account" created

clusterrolebinding "spinnaker-role-binding" created

c. Create accounts need for Helm later

kubectl -n kube-system create sa tiller

--Output will look similar to this

serviceaccount "tiller" created

kubectl create clusterrolebinding tiller --clusterrole cluster-admin

--serviceaccount=kube-system:tiller

--Output will look similar to this

clusterrolebinding "tiller" created

d. Create Spinnaker namespace

kubectl create namespace spinnaker

namespace "spinnaker" created

4. Configure Spinnaker

a. Deployment to host Halyard

kubectl create deployment hal --image

gcr.io/spinnaker-marketplace/halyard:1.35.3

--Output will look similar to this

deployment "hal" created

b. Edit the hal deployment to use the new spinnaker account

kubectl edit deploy hal

c. You want to add the **serviceAccountName** to the **spec** just **above the containers:**

```
...
   spec:
    serviceAccountName: spinnaker-service-account
    containers:
    - image: gcr.io/spinnaker-marketplace/halyard:stable
      imagePullPolicy: IfNotPresent
      name: halyard
      resources: {}
…
```

--Output will look similar to this

deployment "hal" edited

d. Get name of pod

kubectl get pods

Output will look similar to this

---NAME READY STATUS RESTARTS AGE

hal-%%%%% 0/1 ContainerCreating 0 23s

e. Setup bash within the container

kubectl exec -it <CONTAINER-NAME> -- bash

f. cd to the spinnaker user's home directory

g. Configure kubectl

kubectl config set-cluster default --server=https://kubernetes.default
--certificate-authority=/var/run/secrets/kubernetes.io/serviceaccount/ca.crt
kubectl config set-context default --cluster=default
token=$(cat /var/run/secrets/kubernetes.io/serviceaccount/token)
kubectl config set-credentials user --token=$token
kubectl config set-context default --user=user
kubectl config use-context default

h. Instal Helm

curl
https://raw.githubusercontent.com/kubernetes/helm/master/scripts/get > get_helm.sh

--Output will look similar to this

% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 6689 100 6689 0 0 58819 0 --:--:-- --:--:-- --:--:-- 59194

i. Script Updates

sed -i 's/\/usr\/local\/bin/\/home\/spinnaker/g' get_helm.sh
sed -i 's/sudo //g' get_helm.sh
export PATH=/home/spinnaker:$PATH

j. Run the script

chmod 700 get_helm.sh
./get_helm.sh

--Output will look similar to this

Downloading
https://kubernetes-helm.storage.googleapis.com/helm-v2.8.2-linux-amd64.tar.gz
Preparing to install into /usr/local/bin
helm installed into /usr/local/bin/helm
Run 'helm init' to configure helm.

k. Run Helm against cluster

helm init --service-account tiller --upgrade

--Output will look similar to this

Creating /root/.helm
Creating /root/.helm/repository
Creating /root/.helm/repository/cache
Creating /root/.helm/repository/local
Creating /root/.helm/plugins
Creating /root/.helm/starters
Creating /root/.helm/cache/archive
Creating /root/.helm/repository/repositories.yaml
Adding stable repo with URL: https://kubernetes-charts.storage.googleapis.com
Adding local repo with URL: http://127.0.0.1:8879/charts
$HELM_HOME has been configured at /root/.helm.

Tiller (the Helm server-side component) has been installed into your Kubernetes
Cluster.

Please note: by default, Tiller is deployed with an insecure 'allow unauthenticated users' policy.
For more information on securing your installation see:
https://docs.helm.sh/usi
ng_helm/#securing-your-helm-installation

5. Config Spinnaker Deployment
   a. Start by setting up the Docker registry
      i. Set the registry address

ADDRESS= <ADDRESS>
REGION=<REGION>

      ii. Enable the provider

hal config provider docker-registry enable

      iii. Set up authentication

apt install python3-pip
pip3 install awscli

The next command could be done manually with access to the AWS console

aws iam attach-role-policy --policy-arn arn:aws:iam::aws:policy/AmazonEC2ContainerRegistryReadOnly --role-name SpinnakerInstanceRole

      iv. Add the account

**hal config provider docker-registry account add my-ecr-registry **
**--address $ADDRESS **
--username AWS
--password-command "aws --region $REGION ecr get-authorization-token --output text --query 'authorizationData[].authorizationToken' | base64 -d | sed 's/^AWS://'"

b. Set up Storage Settings //Commands may differ if buckets already exist

**hal config storage s3 edit **
**--access-key-id $YOUR_ACCESS_KEY_ID **
**--secret-access-key **
--region $REGION
hal config storage edit --type s3

c. Set up to use Kubernetes

hal config provider kubernetes enable
hal config provider kubernetes account add my-k8s-account --docker-registries my-docker-registry
hal config deploy edit --type distributed --account-name my-k8s-account

d. Define Version

i. See the available versions

1. **hal version list**

ii. Pick one

hal config version edit --version <version you picked>
--Output will look similar to this
+ Get current deployment
Success
+ Edit Spinnaker version
Success
+ Spinnaker has been configured to update/install version "1.20.4".
Deploy this version of Spinnaker with `hal deploy apply`.

6. Create services to front UI and API

a. Create spinsvcs.yaml

apiVersion: v1
kind: Service
metadata:
namespace: spinnaker
labels:

```
      app: spin
      stack: gate
  name: spin-gate-public
spec:
  type: LoadBalancer
  ports:
  - name: http
    port: 8084
    protocol: TCP
 selector:
   load-balancer-spin-gate: "true"
---
apiVersion: v1
kind: Service
metadata:
  namespace: spinnaker
  labels:
    app: spin
    stack: deck
  name: spin-deck-public
spec:
  type: LoadBalancer
  ports:
  - name: http
    port: 9000
    protocol: TCP
 selector:
    load-balancer-spin-deck: "true"
```

b. Create services

kubectl create -f spinsvcs.yaml

--Output will look similar to this

service "spin-gate-np" created

service "spin-deck-np" created

c. Alternatively, create these services imperatively:

i. export NAMESPACE=spinnaker

ii. $ kubectl expose service -n ${NAMESPACE} spin-gate --type LoadBalancer \

--port 80 \ #use port 80 if overriding base url to DNS, so you can visit <DNS-deck> without specifying a port

--target-port 8084 \

--name spin-gate-public

iii. $ kubectl expose service -n ${NAMESPACE} spin-deck --type LoadBalancer \
--port 80 \ #use port 80 if overriding base url to DNS, so you can visit spin-deck.revaturelab.com without specifying a port
--target-port 9000 \
--name spin-deck-public

d. (not currently working) Send UI and API to DNS addresses, and update CORS
   i. hal config security ui edit --override-base-url <DNS-deck >
   ii. hal config security api edit --override-base-url <DNS-gate >
   iii. hal config security api edit --cors-access-pattern <DNS-deck >

7. Enable and configure the artifacts you want to use
8. Apply the configuration
   a. **hal deploy apply**
   b. UI will be reachable at <DNS-deck>

References:

"How to deploy Spinnaker on Kubernetes: a quick and dirty guide"
https://www.mirantis.com/blog/how-to-deploy-spinnaker-on-kubernetes-a-quick-and-dirty-guide/
"Docker Registry - Spinnaker"
https://www.spinnaker.io/setup/install/providers/docker-registry/
"S3 - Spinnaker"
https://www.spinnaker.io/setup/install/storage/s3/
"Exposing Spinnaker | Armory"
https://kb.armory.io/admin/expose-spinnaker/